Final Exam for Real Time Systems

2006 Oct 20, 15-20

Wang Yi

Instructions:

1. You may use a mini-calculator (not a computer!) and a dictionary.
2. Do NOT write on the back side.
3. Put page number on each page.
4. You may write in English or Swedish.
5. State which problems you have solved in the following table.
6. Please handle in this coverage page together with your solutions.

Problem	Solved	Max. Points	Your Points
1		30	
2		10	
3		10	
4		10	
5		10	
6		10	
7		10	
8		10	
	SUMMA:	100	

Name : ....

Pers.no. : ....

Problem 1 *(30p)*

1. *Explain the semantics of* **delay 20; x:=10** *and* **delay until 20; x:=10** *in Ada.*

2. *Describe briefly the concept of "Synchronous Hypothesis".*

3. *What is the essential difference between RTOS and general-purpose OS?*

4. *Explain briefly the concepts: "Multicore", "Multithreading processor", "Multitasking" and "Multiprocessor".*

5. *Explain briefly how the "message scheduling mechanism" (i.e. the arbitration mechanism) of CAN works.*

6. *What is the difference between DMS and EDF? Are they optimal? If yes, explain in what sense.*

7. *Describe two reasons why modeling and verification techniques may help in developing real-time systems.*

8. *Give an example to explain why non-preemptive EDF and SJF are not optimal in scheduling tasks with deadlines.*

9. *Are RMS and EDF "stable"? Explain your answers.*

10. *Describe one method to reduce memory usage for a set of soft RT tasks running on a priority-based RTOS.*

Problem 2 *(10p) Assume a system containing three periodic tasks:*

$$\{(20, 100), (40, 150), (20, T3)\}$$

with worst case execution times 20, 40 and 20 and task periods: 100, 150 and T3. Now assume that the instances of the third task always arrive at the same time as the second task, but it may not be possible to start before the completion of the second task.

What are the period, release jitter and worst case response time of the third task?

Problem 3 *(10p) A watch-dog is monitoring the worst-case response time of a task. If the response time of the task exceeds a given limit D (deadline), the watch-dog should set a variable WARNING to 0, and both the watch-dog and the task should switch to a Unsafe-node at the same time.*

- *Draw two timed automata to model the behaviour of the watch-dog and the task.*
- *Write a requirement formula in TIMES to check the Unsafe-node of the watch-dog is reachable. Add timing constraints to the task automaton such that the Unsafe-node of the Watch-dog is not reachable.*

Problem 4 *(10p)*

1. *Describe two methods to improve the average response times for soft RT tasks.*
2. *Give a method to handle non-periodic hard RT tasks for a system running RMS. Describe the equation to calculate the worst case response time for a non-periodic hard RT task.*

Problem 5 *(10p) Study the following Ada-like program. Model the two tasks using a network of two timed automata. Can the automaton for task B reach node STOP when it is running together with the automaton for task A?*

Task body A is	*Task body B is*
	loop
	select
	begin
	accept Call;
loop	*goto foo*
next := clock;	*end*
delay until next+100;	*or*
if $x = 1$ *then B.Call*	*begin*
else $x := 1$	*delay 20;*
end loop	$x := 0$;
	end
	end select
	end loop
	foo: STOP

Problem 6 *(10p)*

1. *Describe the un-bounded priority inversion problem.*

2. *Describe the resource access protocols: BIP (Basic Priority Inheritance Protocol) and HLP (High Locker's Priority Protocols). Please use examples if needed.*

3. *Can BIP and HLP prevent deadlocks? Motivate your answer.*

4. *The two standard operations P and V on semaphores are implemented according to the following pseudo-code:*

P(S)	V(S)

Disable-interrupt;
if S.counter > 0
then S.counter -- 1;
else {
insert(current-task, S.queue);
schedule()};
Enable-interrupt
	Disable-interrupt;
	If non-empty(S.queue)
	then {
	next-to-run := get-first(S.queue);
	insert(next-to-run, Ready-queue);
	schedule()}
	else S.counter ++ 1;
	Enable-interrupt

Modify the above code to implement HLP. You should also describe what information should be kept in the TCB (task control block) and SCB (semaphore control block) for your implementation.

Problem 7 *(10p) Given a set of periodic tasks some of which may have the same periods.*

1. *Describe how RMS works for the task set.*

2. *Describe how the RMS sufficient schedulability test (i.e. using the utilization bound) works.*

3. *Describe how to calculate the worst case response times for tasks.*

4. *Give a necessary utilization bound for the schedulability of the task set if a 10-processor system is used to compute the tasks (ignore the overheads).*

Problem 8 *(10p) Assume a CAN bus running at 1Mbits per second, connecting three stations (nodes) A, B, and C.*

1. *On node A, there are two tasks. One is sending a message with identity 5 at most every 50ms and The other is sending a message with identity 7 at most every 60ms.*
2. *On node B, there are two tasks. One is sending a message with identity 8 at most every 100ms and the other is sending a message with identity 2 at most every 10ms.*
3. *On node C, there is a single task sending a message with identity 2 at most every 20ms.*

The transmitted messages are of fixed size (100 bits each). Assume that the CAN controllers have sufficient buffer capacity, no transmission errors, and no jitters. What is the worst case transmission delay (i.e. time from queuing to completed message transmission) for the messages with identity 7? Motivate your answer.